*Journal de Théorie des Nombres*
*de Bordeaux* **6** *(1994), 21–38*

# Efficient computation of addition chains

by F. Bergeron†, J. Berstel‡ and S. Brlek†

Abstract – The aim of this paper is to present a unifying approach to the computation of short addition chains. Our method is based upon continued fraction expansions. Most of the popular methods for the generation of addition chains, such as the binary method, the factor method, etc..., fit in our framework. However, we present new and better algorithms.

We give a general upper bound for the complexity of continued fraction methods, as a function of a chosen strategy, thus the total number of operations required for the generation of an addition chain for all integers up to $n$ is shown to be $O(n \log^2 n \gamma_n)$, where $\gamma_n$ is the complexity of computing the set of choices corresponding to the strategy. We also prove an analog of the Scholz-Brauer conjecture.

## 1. Introduction

An *addition chain* for a positive integer $n$ is a sequence of positive integers $\mathcal{C} = (n_0, n_1, \dots, n_s)$ such that

(i) $n_0 = 1$ and $n_s = n$,

(ii) for each $i$, $1 \le i \le s$, there exist $k, j < i$ such that $n_i = n_j + n_k$.

The integer $s$ is the *length* of the chain $\mathcal{C}$ and is denoted by $|\mathcal{C}|$. The *chain length* $\ell(n)$ of $n$ is the minimal length of all possible chains for $n$. Clearly, this notion extends to sets of numbers as well, and we shall denote the chain length of $\{m_1, \dots, m_t\}$ by $\ell(m_1, \dots, m_t)$.

Brauer (in [1]) introduced a special class of addition chains, namely *star-chains*, characterized by the condition that

(ii′) for each $i$, $1 \le i \le s$, there exists $k < i$ such that $n_i = n_{i-1} + n_k$.

These chains play a major role in our study.

Manuscrit reçu le 17 juin 1992.
†With partial support from NSERC-Canada and FCAR-Québec.
‡With the support of PRC "Mathématiques et Informatique" and ESPRIT, BRA working group 6317–ASMICS 2.

Addition chains for $n$ generate multiplication schemes for the computation of $x^n$. For instance, the chain $(1, 2, 4, 8, 9, 17, 34, 43)$ leads to the following scheme for the computation of $x^{43}$:

$$xx = x^2,\ x^2x^2 = x^4,\ x^4x^4 = x^8,\ xx^8 = x^9,$$
$$x^8x^9 = x^{17},\ x^{17}x^{17} = x^{34},\ x^9x^{34} = x^{43}.$$

Clearly this establishes a correspondance between addition chains and such schemes. Furthermore, the length of an addition chain for $n$ is equal to the corresponding number of multiplications required for the computation of $x^n$. Thus, the smallest number of such multiplications is given by the chain length $\ell(n)$ of $n$.

This subject has a long history, a detailed account of which is given by Knuth in his second volume [9]. Any explicit algorithm for the generation of addition chains clearly sets an upper bound on the function $\ell(n)$. Thus the usual binary expansion algorithm (see [9]) implies that $\ell(n) \leq \lambda(n) + \nu(n) - 1$, where $\lambda(n) = \lfloor \log_2(n) \rfloor$ and $\nu(n)$ is the number of 1 in the binary expansion of $n$. However, the problem of computing the exact value of $\ell(n)$ seems to be difficult. Indeed, a slightly more complex problem, namely the problem of computing the chain length for a set of integers, has been shown to be $NP$-complete [7]. Therefore, it is interesting to consider sub-optimal addition chains, provided that they can be constructed in an efficient way.

In a previous paper [3] (see also the work of Semba [11]), we have introduced such a class of sub-optimal addition chains for positive integers $n$. These were obtained through continued fraction expansions for $n/k$, where $k$ is some integer chosen between 2 and $n-1$. We call chains of this form *continued fraction addition chains*, or simply *cf-chains*. We proved in this same paper that the Scholz-Brauer conjecture holds for cf-chains. This result implies that for an infinite class of integers, cf-chains are much closer to optimal addition chains than the chains obtained by the usual binary method. A precise form of this assertion can be found in paragraph 5. Moreover we show (Theorem 2) that the length of cf-chains satisfies the same asymptotic bound as $\ell(n)$

$$\lim_{n\to\infty} \frac{L(\{n\} \cup \tau(n), \sigma)}{\lambda(n)} = 1.$$

Even though minimal-length cf-chains are not optimal, they have the nice property of being easy to compute and significantly shorter on the average than chains obtained by the binary method. This is clearly important when the cost of even one multiplication is high. Moreover, most of the popular effective strategies for computing addition chains are obtained as special cases of the cf-chain method. Thus, minimal length cf-chains will systematically be shorter than the chains obtained by these other methods. We further prove that the total number of operations required for the generation of an addition chain for all integers up to $n$ is $O(n \log^2 n \gamma_n)$, where $\gamma_n$ is the complexity of computing the set of choices corresponding to the strategy. This takes into account the complexity of arithmetics with multiple-precision integers. Thus in the case where $\gamma_n = O(\log n)$, one gets the upper bound $O(n \log^3 n)$. This holds for the dyadic strategy.

## 2. Continued fraction chains

Let us introduce two simple operations on addition chains. Given two addition chains $\mathcal{C} = (n_0, n_1, \dots, n_s)$ for $n$ and $\mathcal{C}' = (m_0, m_1, \dots, m_t)$ for $m$, define the product $\mathcal{C} \otimes \mathcal{C}'$ to be

$$\mathcal{C} \otimes \mathcal{C}' = (n_0, n_1, \dots, n_s, n\, m_1, n\, m_2, \dots, n\, m_t).$$

It is clear that this chain for $nm$ is of length $|\mathcal{C}\mathcal{C}'| = |\mathcal{C}| + |\mathcal{C}'|$. Now, if $\mathcal{C} = (n_0, n_1, \dots, n_s)$ is a chain for $n$, and $j$ is one of the integers appearing in $\mathcal{C}$, then define $\mathcal{C} \oplus j$ to be

$$\mathcal{C} \oplus j = (n_0, n_1, \dots, n_s, n_s + j).$$

Obviously the length of this chain is $|\mathcal{C} \oplus j| = |\mathcal{C}| + 1$.

On the other hand, recall that for an integer $n$ and any $k$ belonging to $\{2, 3, \dots, n-1\}$, the continued fraction expansion of $n/k$ is

$$\frac{n}{k} = u_r + \cfrac{1}{u_{r-1} + \cfrac{1}{\ddots + \cfrac{1}{u_2 + \cfrac{1}{u_1}}}}.$$

We shall denote this continued fraction by $[u_1, u_2, \dots, u_r]$. The *semi-continuants* $Q_i$ of this continued fraction are:

$$Q_0 = \gcd(n, k), \quad Q_1 = Q_0 u_1,$$
$$Q_i = Q_{i-1} u_i + Q_{i-2}, \text{ for } 2 \le i \le r.$$

Observe that by construction $Q_r = n$.

A similar recursion, also derived from the continued fraction expansion of $n/k$, yields a *continued fraction addition chain*, cf-chain for short. Let $\mathcal{C}(d)$ be some cf-chain for $d = \gcd(n,k)$; and for each $i$, $1 \le i \le r$, let $\mathcal{C}_i = \mathcal{C}(u_i)$ be some cf-chain for $u_i$ . We define the sequence of cf-chains $\mathcal{X}_i$ by:

$$\mathcal{X}_0 = \mathcal{C}(d), \quad \mathcal{X}_1 = \mathcal{X}_0 \otimes \mathcal{C}_1,$$
$$\mathcal{X}_i = (\mathcal{X}_{i-1} \otimes \mathcal{C}_i) \oplus Q_{i-2}, \text{ for } 2 \le i \le r.$$

Hence $\mathcal{X}_i$ is a chain for $Q_i$, thus $\mathcal{X}_r$ is also a chain for $n$ and we shall denote it $\mathcal{C}(n)$. All chains obtained in this manner will be called *cf-chains*. Since for each $i \ge 2$, $|\mathcal{X}_i| = |\mathcal{X}_{i-1}| + |\mathcal{C}_i| + 1$, and $|\mathcal{X}_1| = |\mathcal{C}_1| + |\mathcal{C}(d)|$, we have

$$|\mathcal{C}(n)| = |\mathcal{C}(d)| + r - 1 + \sum_{i=1}^{r} |\mathcal{C}_i|.$$

Clearly, the construction of a chain $\mathcal{C}(n)$ for some integer $n$ depends on the choice of $k$ as well as on the choice of the cf-chains $\mathcal{C}(d), \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r$. The resulting chain will vary according to the choices made. In any case, the chain for $2^a$ should always be $(1, 2, 4, \ldots, 2^a)$, because it is clearly of minimal length.

Observe that every cf-chain is also a star-chain in Brauer's terminology. This implies that cf-chains are not always optimal since there exist integers for which no optimal chain is a star-chain (12509 is the first integer of this kind [9]). Conversely, there are star-chains which are not cf-chains. For $n = 367$, a computer program showed [3] that the shortest cf-chain has length 12 whereas a shortest star-chain has length 11. However, the advantage of cf-chains over arbitrary star-chains is that they are easier to compute. This is precisely formulated in Theorem 1 below.

**Example 1**. For $n = 86$, let us choose $k = 10$. The corresponding continued fraction is $[2, 1, 1, 8]$, $d = \gcd(86, 10) = 2$, and $(Q_0, Q_1, Q_2, Q_3, Q_4) = (2, 4, 6, 10, 86)$. Hence the cf-chain produced by our method (provided we choose the chain $(1, 2, 4, ..., 2^a)$ for $2^a$) is

$(1, 2) \otimes (1, 2) = (1, 2, 4)$,

$(1, 2, 4) \oplus 2 = (1, 2, 4, 6)$,

$(1, 2, 4, 6) \oplus 4 = (1, 2, 4, 6, 10)$,

$(1, 2, 4, 6, 10) \otimes (1, 2, 4, 8) = (1, 2, 4, 6, 10, 20, 40, 80)$,

$$(1,2,4,6,10,20,40,80) \oplus 6 = (1,2,4,6,10,20,40,80,86).$$

Observe that this chain has minimal length 8, whereas the binary method produces the chain $(1,2,4,5,10,20,21,42,43,86)$ which has length 9.

## 3. Strategies

As we have already mentioned, we need to specify how to choose the auxiliary integer $k \in \{2,3,\ldots,n-1\}$ that will be used for the continued fraction expansion. Thus we shall define a *strategy* to be a function $\gamma$ that determines for each integer $n$ (which is not a power of 2) some non-empty subset $\gamma(n)$ of $\{2,3,\ldots,n-1\}$. For $2^a$, we simply set the minimal length cf-chain to $(1,2,4,...,2^a)$. We can now give a precise form to the method as an algorithm parametrized by a strategy $\gamma$.

**Algorithm** *Minchain*$(n,\gamma)$

(*produces a shortest cf-chain for* $n$ *according to* $\gamma$)

**if** $n = 2^a$ **then** *return* $(1,2,4,\ldots,2^a)$

**elif** $n = 3$ **then** *return* $(1,2,3)$

**else** *choose some* $k \in \gamma(n)$ *such that* *Chain*$(\{n,k\},\gamma)$

*has minimal length and return the chain*

**endif**

**end** *Minchain.*

where

**Algorithm** *Chain*$(\{n_1,n_2,\ldots,n_k\},\gamma)$

(*produces a cf-chain for* $\{n_1,n_2,\ldots,n_k\}$)

**if** $n_2 \leq 1$ **then** *return* *Minchain*$(n_1,\gamma)$

**else let** $q = (n_1$ **div** $n_2)$**;** $r = (n_1$ **rem** $n_2)$

**if** $r = 0$ **then** *return* *Chain*$(\{n_2,n_3,\ldots,n_k\},\gamma) \otimes$ *Minchain*$(q,\gamma)$

**else** *return* (*Chain*$(\{n_2,n_3,\ldots,n_k,r\},\gamma) \otimes$ *Minchain*$(q,\gamma)) \oplus r$

**endif**

**endif**

**end***Chain.*

The length of the chain for $\{n_1, n_2, \ldots, n_k\}$, $(n_i > n_{i+1})$, produced by this algorithm will be denoted by $L(\{n_1, n_2, \ldots, n_k\}, \gamma)$. Now if we further denote by $\ell(n, \gamma)$ the length of the chain for $n$ produced by algorithm *Minchain*, then one has the following identities (see [3, 4])

$$\ell(n,\gamma) = \begin{cases} a, \text{ if } n = 2^a; \\ 2, \text{ if } n = 3; \\ \min\limits_{k \in \gamma(n)} L(\{n,k\},\gamma), \text{ otherwise} \end{cases} \tag{1}$$

and

$$\begin{aligned} &L(\{n_1, n_2, \ldots, n_k\}, \gamma) = \\ &\qquad \begin{cases} L(\{n_2, n_3, \ldots, n_k\}, \gamma) + \ell(q,\gamma), \text{ if } r = 0; \\ L(\{n_2, n_3, \ldots, n_k\}, \gamma) + \ell(q,\gamma) + 1, \text{ if } r = 1, 2; \\ L(\{n_2, n_3, \ldots, n_k, r\}, \gamma) + \ell(q,\gamma) + 1, \text{ otherwise}, \end{cases} \end{aligned} \tag{2}$$

where $n_1 = q n_2 + r$, with $0 \le r < n_2$. In the sequel of this paper, these functions will play an important role in the description of the properties of the above algorithm.

The following recursive expression for the length of the cf-chain produced for $n$ follows from these definitions. Denote by $\ell(n, \gamma)$ the length of a shortest cf-chain for $n$ according to the strategy $\gamma$, and by $L(\{n, k\}, \gamma)$ the length of a shortest cf-chain for $n$ containing $k$ and obtained through the continued fraction expansion of $n/k$. Then $\ell(1, \gamma) = 0$, and

$$\begin{aligned} &L(\{n,k\},\gamma) = \\ &\qquad \begin{cases} \ell(k,\gamma) + \ell(q,\gamma), \text{ if } n = kq; \\ L(\{k,r\},\gamma) + \ell(q,\gamma) + 1, \text{ if } n = kq + r, \text{ where } 0 < r < k. \end{cases} \end{aligned} \tag{2'}$$

Many interesting strategies can be chosen for the generation of cf-chains. For each of these strategies, say $\gamma$, we shall consider the complexity of computing a $\gamma$-*chain*, i.e. a cf-chain obtained with strategy $\gamma$.

**Binary strategies**, [9]. The most popular strategy is obtained by choosing for all $n$,

$$\beta(n) = \left\{ \left\lfloor \frac{n}{2} \right\rfloor \right\}.$$

This is exactly the *binary method*, in Knuth's terminology. For instance, the cf-chain for 87 obtained with this strategy is:

$$(1, 2, 4, 5, 10, 20, 21, 42, 43, 86, 87).$$

The length of a binary chain is known to be $\ell(n, \beta) = \lambda(n) + \nu(n) - 1$. In fact, this value also follows from recurrence (2) with $\gamma = \beta$. One could also consider the *co-binary* strategy $\alpha$:

$$\alpha(n) = \{(n + 1) \textbf{ div } 2\}.$$

The cf-chain for 87 obtained with the co-binary strategy, is:

$$(1, 2, 3, 5, 10, 11, 21, 22, 43, 44, 87).$$

**Factor strategy**, [9]. The *factor* method (Knuth) corresponds to the strategy:

$$\pi(n) = \begin{cases} \{n-1\}, & \text{if } n \text{ is prime;} \\ \{n-1, q\}, & \text{otherwise, where } q \text{ is the smallest prime dividing } n. \end{cases}$$

In this case, even though the set of candidates is small, the computation of $\ell(n, \pi)$ is clearly equivalent to the factorization of $n$. Therefore, the efficiency of this method is rather bad. A factor chain for 87 is:

$$(1, 2, 3, 6, 12, 24, 48, 72, 84, 87).$$

**Total strategy**, [2, 3, 11]. The *total* strategy $\theta$ corresponds to the choice of all acceptable candidates, i.e.:

$$\theta(n) = \{2, 3, \ldots, n - 1\}.$$

As we shall see below, the complexity of the corresponding algorithm is $O(n^2 \log^2 n)$. One minimal total chain for 87 is obtained with $k = 17$ :

$$(1, 2, 4, 8, 16, 17, 34, 68, 85, 87).$$

This is an optimal chain for 87. However, it is not always true that a minimal cf-chain is a shortest star-chain. For example, a shortest star-chain for 367 has length 11, whereas the total strategy yields $\ell(367, \theta) = 12$ (see [3]).

**Dyadic strategy**, [2, 3]. Because of this relatively high complexity, it is interesting to introduce faster near-optimal methods. One such is the *dyadic* strategy:

$$\delta(n) = \left\{ \left\lfloor \frac{n}{2^j} \right\rfloor \;\mid\; j = 1, \ldots, \lambda(n) - 1 \right\}.$$

The complexity of this strategy is significantly smaller, namely $O(n \log^3 n)$. Observe that $\beta(n) \subset \delta(n)$. Therefore minimal dyadic chains are always shorter than binary chains, that is $\ell(n, \delta) \leq \ell(n, \beta)$ for all $n$. For our running example, a minimal dyadic chain is:

$$(1, 2, 4, 6, 10, 20, 40, 80, 86, 87).$$

This chain is also an optimal chain for 87.

**Fermat's strategy**. A strategy which is much faster to compute is *Fermat*'s strategy:

$$f(n) = \left\{ \left\lfloor \frac{n}{2^{2^j}} \right\rfloor \;\mid\; j = 0, \ldots, \lambda(\lambda(n) - 1) \right\}.$$

It yields the following (optimal) addition chain:

$$(1, 2, 4, 5, 10, 20, 40, 80, 85, 87).$$

The interest of Fermat's strategy lies in the gain of computation time since $f(n)$ is a set of size $O(\log \log n)$. Computer calculations have shown that $\ell(n, f(n)) \leq \ell(n) + 1$ for all $n$ up to 1000.

**Dichotomic strategy**. This strategy is defined by

$$\sigma(n) = \left\{ \left\lfloor \frac{n}{2^{\lceil \lambda(n)/2 \rceil}} \right\rfloor \right\}.$$

Since the set of choices here is a singleton, the dichotomic strategy is *deterministic* (like the binary strategy). Consequently, both the chain and the chain length are computed without any backtrack. The length of the chains produced are logarithmic in the argument, but the chains are not always optimal with these fast strategies. As shown in Theorem 3 below, the dichotomic strategy provides much shorter chains than the binary one for a large family of integers.

The data reported in Table 1 has been obtained with MAPLE© and a program written in C++ running on a MIPS2000. For each strategy $\gamma$, the table lists the values

$$\sum_{n=2}^{N} \ell(n,\gamma).$$

The values for $\ell(n)$ were taken from Knuth [9]. The time shown is the time required for computing a table of all chains for integers up to 1000.

| Strategy | Total length ($N$=1000) | Difference with $\ell$ | Time (sec.) Maple | Time (min.) C-program |
|---|---|---|---|---|
| Binary | 11925 | 1117 | 8.7 | $<<$ 0.01 |
| Factor | 11088 | 280 | 17.2 | |
| Dichotomic | 11064 | 256 | 14.4 | $<<$ 0.01 |
| Fermat | 10927 | 119 | 17.7 | 0.03 |
| Dyadic | 10837 | 29 | 24.3 | 0.09 |
| Total | 10821 | 13 | 2764.3 | 15.15 |
| Optimal | 10808 | 0 | | 343.00 |

Table 1. *Comparison of the strategies.*

**Remarks.** The table shows that on the average, the binary (schoolbook) method is by far the *worst* with respect to the length. The computation for the factor method does not really take into account the complexity of computing the prime factorizations involved. Indeed, MAPLE factorization procedure has a built-in table of small primes (up to 1000). We have not produced a similar table to be used with our C++ program in order to compare it but the time obtained with the MAPLE program gives a clear indication of the projected time. Also, the reason for the missing data about the *Optimal* strategy is clear.

## 4. Results

Each strategy defines an instantiation of the general algorithms for the generation of cf-chains. Although the complexity of these algorithms clearly depends on the complexity of computing the strategy $\gamma$, we are mainly interested in evaluating the functions $\ell$ and $L$. Indeed, the effective construction of an optimal cf-chain with respect to a strategy $\gamma$, is similar to the computation of its length. More precisely, addition chains are represented by linked lists. Each node represents one term of the chain, and two pointers link this node to the terms whose sum give this term. We show in Fig. 1 an example of such a representation for the chain $(1, 2, 4, 5, 10, 20, 40, 80, 85, 87)$.

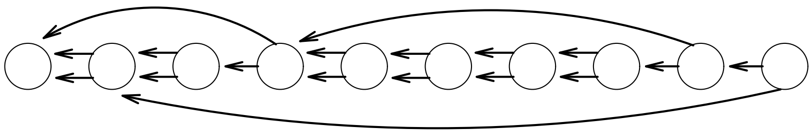

Fig. 1. *List representation of* $(1, 2, 4, 5, 10, 20, 40, 80, 85, 87)$.

Observe that the terms of the chain are not part of this representation. This allows for a constant time implementation of the basic operations $\otimes$ and $\oplus$. So the construction of optimal cf-chains reduces in linear time to the computation of $\ell$ and $L$ as defined by (1) and (2′).

The complexity of computing $\ell(n, \gamma)$ is a function of $n$, of the size of $\gamma(n)$ and of the complexity of computing $\gamma(n)$. Let $\gamma_n$ denote the complexity of computing $\gamma(n)$, then we have the following result.

THEOREM 1. *Assume that $\gamma_n$ is an increasing function of $n$, then the complexity of computing the function $\ell$ for all integers up to $n$ is $O(n\gamma_n \log^2 n)$, if one takes into account the complexity of doing arithmetic with multiple-precision integers.*

*Proof.* We assume that all the $\ell(m, \gamma)$, for $m < n$, have already been computed and stored in an array. Thus the computation of $L(\{n, k\}, \gamma)$ using (1) and (2) has the complexity of producing the continued fraction expansion of $n/k$. For $k$ in $\gamma(n)$, the complexity of expanding $n/k$ as a continued fraction (with multiple-precision integers) has been shown by G. E. Collins [6] to be $O(\log(k)(1+\log(n/d)))$, where $d = \gcd(n, k)$. Hence, the complexity of computing $L(\{n, k\}, \gamma)$ is $O(\log^2 n)$ and the complexity of computing $\ell(n, \gamma)$ out of these is $O(\gamma_n \log^2 n)$. Thus the total complexity we are looking for is $\sum_{j=2}^{n} \gamma_j \log^2 j$, which is clearly in $O(n\gamma_n \log^2 n)$. ∎

In practice, we have the following expressions for the complexities

$$\beta_n = O(1), \quad \theta_n = O(n), \quad \delta_n = O(\log n), \quad f_n = O(\log\log n).$$

It is straightforward to show that if $\gamma(n) \subset \gamma'(n)$ for two strategies $\gamma$ and $\gamma'$, then $\ell(n,\gamma') \le \ell(n,\gamma)$. In particular, $\ell(n,\theta) \le \ell(n,\gamma)$ where $\theta$ is the total strategy. The diagram in Fig. 2 describes the relations between the strategies above, where an arrow $\gamma \longrightarrow \gamma'$ between two strategies $\gamma$ and $\gamma'$ indicates the relation $\gamma'(n) \subset \gamma(n)$.

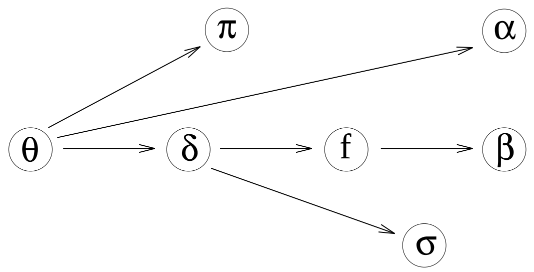


Fig. 2. *The hierarchy of strategies.*

## 5. Computing short chains

We briefly recall the major results about the length of addition chains. Schönhage [10] established the lower bound

$$\ell(n) \ge \log_2(n) + \log_2(\nu(n)) - 2.13.$$

Brauer [1] determined the asymptotic behavior of $\ell(n)$ by producing the upper bound,

$$\ell(n) \le \lambda(n) + \lambda(n)/p + 2^p - 2$$

for all $p \ge 1$. For a suitable $p$, (for instance $p = \lfloor 1/2 \log_2(\lambda(n)) \rfloor$), we get

$$\lim_{n\to\infty} \frac{\ell(n)}{\lambda(n)} = 1.$$

Thurber [14] improved Brauer's result as follows: take first the binary representation of $n$; set $m = \lambda(n)+1$; then, starting from the leading digit, partition the binary word into equal parts of length $p$, producing the set $\{n_1, n_2, ..., n_{\lceil m/p \rceil}\}$. Then, each $n_i$ is in the initial set $\{1, 2, 3, ..., 2^p - 1\}$, and the number $n$ is produced by applying the rules:

R1) $M_1 = n_1$;

R2) $M_i = M_{i-1}2^p + n_i$, for $i = 2, ..., \lceil m/p \rceil - 1$;

R3) $M_{\lceil m/p \rceil} = n = M_{\lceil m/p \rceil - 1}2^{m-p(\lceil m/p \rceil - 1)} + n_{\lceil m/p \rceil}$.

Note that multiplication by $2^p$ is achieved by shifting, and if $n_i$ is even, then it can be replaced by an odd number $n_i'$ such that $n_i = n_i' 2^j$ for some $j \geq 1$. This does not affect the total number of shifts. Therefore, $\ell(n)$ is bounded by the total number of operations needed to produce $n$, namely

$$\ell(n) \leq \lambda(n) - (p-1) + \lfloor \lambda(n)/p \rfloor + 2^{(p-1)},$$

where $2^{(p-1)}$ stands for the computation of all odd numbers less than $2^p$. In that same paper, Thurber pointed out that this construction could be improved for small values of $n$: he proposed to take a non-uniform partition of the binary representation of $n$, based on an analysis of the binary pattern. Recently, Bos and Coster [5] used this method to produce addition chain heuristics in a more restricted context. They investigated applications to cryptography, namely the computation of powers appearing in the well-known algorithm of Rivest, Shamir and Adleman (RSA). Bos and Coster consider numbers such that $\lambda(n) \leq 512$.

Here, we shall only consider a uniform partition and replace the initial set of $2^{(p-1)}$ odd numbers by a chain for the set $\{n_1, n_2, ..., n_{\lceil m/p \rceil}\}$, yielding the upper bound

$$\ell(n) \leq \lambda(n) - (p-1) + \lfloor \lambda(n)/p \rfloor + \ell(n_1, n_2, ..., n_{\lceil m/p \rceil}).$$

Thus, the problem of computing $\ell(n)$ is reduced to the computation of an optimal chain for the set $\{n_1, n_2, ..., n_{\lceil m/p \rceil}\}$. This suggests the use of Yao's method [15]. Yao's algorithm is asymptotically optimal, but there is still place for improvements when the numbers $n_i$ are small.

Brauer's bound is easily expressed within our context. To do so, we first define the $m$-ary strategy. Let $m = \lambda(n) + 1$, $\Delta = m - p(\lceil m/p \rceil - 1)$, and $t = \lfloor m/p \rfloor - 1$.

**$m$-ary strategy**.

$$\tau(n) = \left\{ \left\lfloor \frac{n}{2^{ip+\Delta}} \right\rfloor \;\middle|\; i = 0, \ldots, t;\; p = \lfloor 1/2 \log_2(\lambda(n)) \rfloor \right\}.$$

The next theorem characterizes the asymptotic behavior of $L(\{n\} \cup \tau(n), \sigma)$. Its proof follows that of Knuth [9] for $\lim_{n\to\infty} \ell(n)/\lambda(n) = 1$.

THEOREM 2. *The length of a cf-chain obtained with the $m$-ary strategy satisfies the asymptotic bound*

$$\lim_{n\to\infty} \frac{L(\{n\}\cup\tau(n),\sigma)}{\lambda(n)} = 1.$$

*Proof.* Let $k_i = \lfloor n/(2^{ip+\Delta})\rfloor$ for $i = 0,\ldots,t$. We first show that

$$L(\{n\}\cup\tau(n),\sigma) \le (1+\frac{1}{p})\lambda(n) - p + 2 + L(\{r'_0, r'_1,\ldots,r'_t\},\sigma),$$

where $\{r'_0, r'_1,\ldots,r'_t\}$ is an ordered set such that $r'_i \le 2^p - 1$.

Set $r_0 = k_t$. Following algorithm *Chain*, write $q_1 = n$ div $k_0$ and $r_1 = n$ rem $k_0$. It follows that $q_1 = 2^\Delta$ and $r_1 \le 2^p - 1$. By definition (2) of $L$ we have

$$L(\{n, k_0,\ldots,k_t\},\sigma) \le L(\{k_0,\ldots,k_{t-1}, r'_0, r'_1\},\sigma) + \ell(2^\Delta,\sigma) + 1.$$

Then, for $i = 2,\ldots t+1$, define

$$\begin{aligned} q_i &= k_{i-2} \text{ div } k_{i-1},\\ r_i &= k_{i-2} \text{ rem } k_{i-1}. \end{aligned}$$

Observe that $r'_i \le 2^p - 1$, and that $q_i = 2^p$ for all $i$ excepted perhaps the last one. Indeed, $\lambda(k_{t-1}) = 2\lambda(k_t) = 2(p-1)$, and $k_t = k_{t-1}$ div $2^p$. Let $n_2 = k_{t-1}$ rem $2^p$. Then we have

$$q_{t+1} = \begin{cases} 2^p & \text{if } k_t > n_2,\\ 2^p+1 & \text{otherwise.}\end{cases}$$

It is easy now to check that $\ell(2^p+1,\sigma) = p+1$. We have thus

$$\begin{aligned} L(\{n,k_0,\ldots,k_t\},\sigma) &\le L(\{k_0,k_1,\ldots,k_{t-1},r'_0,r'_1\},\sigma) + \Delta + 1,\\ &\le L(\{k_1,k_2,\ldots,k_{t-1},r'_0,r'_1,r'_2\},\sigma) + p + 1 + \Delta + 1,\\ &\ldots\\ &\le L(\{r'_0,r'_1,\ldots,r'_{t+1}\},\sigma) + t(p+1) + (\Delta+1) + 1. \end{aligned}$$

Substituting $\Delta$ and $t$, we obtain the result claimed.

Finally, since $r'_i \le 2^p - 1$ for all $i$, we have,

$$L(\{r'_0,r'_1,\ldots,r'_{t+1}\},\sigma) \le 2^p - 2$$

and

$$L(\{n\} \cup \tau(n), \sigma) \leq (1 + \frac{1}{p})\lambda(n) - p + 2^p.$$

The limit follows clearly. ∎

**Remarks.**

1. In the proof of Theorem 2, we made use of the dichotomic strategy $\sigma$ in the computation of $\ell(q_i, \sigma)$ and $L(\{r'_0, r'_1, \dots, r'_{t+1}\}, \sigma)$. Clearly, even the binary strategy is sufficient to get the asymptotic bound. However, we could also apply recursively the strategy $\tau$ if the numbers are large enough.

2. In order to get Thurber's method, one needs to determine the parity of the numbers $n_i$, and find the corresponding non-uniform partition of the binary representation of $n$. Since it can be achieved in $\lambda(n)$ time, it is worth doing it in applications, such as RSA computing, where one needs to compute the same power many times. Finally, one can also define a strategy which produces a non-uniform partition of the binary representation of $n$. Let $P = (p_1, p_2, \dots, p_t)$ be a partition of $m = \lambda(n) + 1$ such that

$$n = \sum_{i=1}^{t-1} n_i 2^{\alpha_i} + n_t, \text{ where } \alpha_i = \sum_{j=i+1}^{t-1} p_j.$$

Then we define the $P$-partition strategy by

$$\rho(n) = \left\{ \left\lfloor \frac{n}{2^{\beta_i}} \right\rfloor \mid i = 1, \dots, t-1; \beta_i = \sum_{j=t-i-1}^{t} p_j \right\}.$$

## 6. The Scholz-Brauer inequalities

One of the most intriguing problems concerning addition chains is the so called Scholz-Brauer conjecture (see [1, 3, 4, 8, 12]):

$$\ell(2^n - 1) \leq n - 1 + \ell(n). \tag{3}$$

In [3], we have proved a similar inequality in the case of the dyadic strategy, namely that

$$\ell(2^n - 1, \delta) \leq n - 1 + \ell(n, \theta). \tag{4}$$

This clearly shows that for the family of integers of the form $2^n - 1$, the dyadic method is noticeably better than the binary method. Indeed, it is well known [9] that $\ell(n,\beta) = \lambda(n) + \nu(n) - 1$, whence $\ell(2^n - 1, \beta) = 2n - 2$. Since $\theta$ is more general than $\beta$, it follows that $\ell(n,\theta) \leq \lambda(n) + \nu(n) - 1$. Thus, (4) implies that

$$\ell(2^n - 1, \delta) \leq n - 2 + \lambda(n) + \nu(n).$$

THEOREM 3. *The following equality holds for the dichotomic strategy*

$$\ell(2^n - 1, \sigma) = n - 2 + \lambda(n) + \nu(n) = n - 1 + \ell(n, \beta).$$

*Proof.* Assume first that $n = 2m$. Then $\sigma(2^n - 1) = 2^m - 1$. Consequently, in view of (1)

$$\ell(2^n - 1, \sigma) = L(\{2^n - 1, 2^m - 1\}, \sigma)$$

and, by (2)

$$\ell(2^n - 1, \sigma) = \ell(2^m - 1, \sigma) + \ell(2^m + 1, \sigma). \tag{5}$$

It is easy to see that $\ell(2^m + 1, \sigma) = 1 + m$. It follows from (5) that

$$\ell(2^{2m} - 1, \sigma) = \ell(2^m - 1, \sigma) + m + 1. \tag{6}$$

Assume now that $n = 2m + 1$. Then $\sigma(2^n - 1) = 2^{m+1} - 1$. Again,

$$\ell(2^n - 1, \sigma) = L(\{2^n - 1, 2^{m+1} - 1\}, \sigma),$$

whence, in view of (2), $\ell(2^n - 1, \sigma) = L(\{2^{m+1} - 1, 2^m - 1\}, \sigma) + \ell(2^m, \sigma) + 1$. Using again (2) one gets $L(\{2^{m+1} - 1, 2^m - 1\}, \sigma) = \ell(2^m - 1, \sigma) + 2$. Thus,

$$\ell(2^{2m+1} - 1, \sigma) = \ell(2^m - 1, \sigma) + m + 3. \tag{7}$$

Equations (6) and (7) show that $\ell(2^n - 1, \sigma) = n - 2 + \lambda(n) + \nu(n)$. ∎

We conjecture that another inequality like that of Scholz-Brauer holds for Fermat's strategy, namely:

$$\ell(2^n - 1, f) \leq n - 2 + \lambda(n) + \nu(n).$$

This last inequality has been checked on a computer for $n$ up to 256. Moreover, Scholz-Brauer like inequalities have been derived for vectorial addition chains (see [4]).

## 7. Open problems

Many interesting questions arise from the work presented above. The algorithms presented are based on continued fraction expansions from which it is expected to find partial solutions to problems about optimal addition chains.

Indeed, each strategy defines a sub-class of the class of star-chains for which it would be interesting to investigate the problems described hereafter. Most of them are existing problems about optimal addition chains but restricted to the class of cf-chains.

*Complexity.* So far we know very little about the chain length obtained by the the different methods: the asymptotic complexity is known from the work of Brauer [1] and improved by Thurber [14]; we also know quite well the case of the binary method.

In particular, we can ask for the following questions.

- Is the total strategy asymptotically optimal?

- What is the worst case complexity for the dichotomic strategy? While an upper bound can be found easily, it would be useful to characterize the worst cases.

- What is the average case complexity for the dichotomic strategy? This amounts to a study of the distribution of the partial quotients. Again an upper bound can be easily computed using the distribution of the partial quotients (see Knuth [9] for instance), but an exact value requires more investigations. It is expected that some improvements could be achieved by using ergodic theory.

*Scholz-Brauer inequalities.* As stated earlier, the Scholz-Brauer inequality holds for star-chains, and also for the more general class of $\ell^0$-chains defined by Hansen (see [9]). We derived two similar inequalities with the total and dichotomic strategies, but the question remains open for other strategies.

*Density.* Given a strategy $\gamma$ the problem is to determine the density of the solution set of the equation

$$\ell(n) = \ell(n, \gamma).$$

*Miscellaneous.* Let $c(r,\gamma) = \min(\ell^{-1}(r,\gamma))$ and $d(r,\gamma) = \mathrm{Card}(\ell^{-1}(r,\gamma))$. It is not known whether these functions are monotonic increasing in the case of optimal addition chains. What can be said about cf-chains?

**Acknowledgements.** This paper is a revised and augmented version of a communication [2] that appeared in the proceedings of the XV Latin American Conference on Informatics, held in Santiago (Chile) in 1989. The authors are also thankful to R. Mallette for his programming expertise in writing C++ programs. He also spent a lot of computer time in order to obtain part of the data presented in paragraph 3. We would like to thank Jeffrey Shallit for pointing out the reference to the paper of Ichiro Semba.

## References

[1] A. Brauer, *On addition chains*, Bull. Amer. Math. Soc. **45** (1939), 736–739.

[2] F. Bergeron, J. Berstel, S. Brlek, *A unifying approach to the generation of addition chains*, Proc. XV Latin American Conf. on Informatics, Santiago, Chile July 10-14 (1989), 29–38.

[3] F. Bergeron, J. Berstel, S. Brlek, C. Duboc, *Addition chains using continued fractions*, J. Algorithms **10** (1989), 403–412.

[4] F. Bergeron, J. Olivos, *Vectorial addition chains using Euclid's algorithm*, Research Report, Dpt. Math., UQAM **105** (1989).

[5] J. Bos, M. Coster, *Addition chain heuristics*, Proceedings of CRYPTO 89.

[6] G. E. Collins, *The computing time of the Euclidian algorithm*, SIAM J. Computing **3** (1974), 1–10.

[7] P. Downey, B. Leong, R. Sethi, *Computing sequences with addition chains*, SIAM J. Computing **10** (1981), 638–646.

[8] A. A. Gioia, M. V. Subbarao, M. Sugunama, *The Scholz-Brauer problem in addition chains*, Duke Math. J. **29** (1962), 481–487.

[9] D. E. Knuth, *The art of computer programming*, vol. 2, Addison-Wesley, 1981.

[10] A. Schönhage, *A lower bound for the length of addition chains*, Theoret. Comput. Sci. **1** (1975), 1–12.

[11] I. Semba, *Systematic method for determining the number of multiplications required to compute $x^m$, where $m$ is a positive integer*, J. Information Proc. **6** (1983), 31–33.

[12] E. G. Thurber, *Addition chains and solutions of $\ell(2n) = \ell(n)$ and $\ell(2^n - 1) = n + \ell(n) - 1$*, Discrete Math. **16** (1976), 279–289.

[13] E. G. Thurber, *The Scholz-Brauer problem on addition chains*, Pacific J. Math. **49** (1973), 229–242.

[14] E. G. Thurber, *On addition chains $\ell(mn) \leq \ell(n) - b$ and lower bounds for $c(r)$*, Duke Math. J. **40** (1973), 907–913.

[14] A. C.-C. Yao, *On the evaluation of powers*, SIAM J. Comp. **9** (1976), 100–103.

F. Bergeron et S. Brlek,
LACIM, Université du Québec à Montréal,
CP 8888 Suc. A, Montréal, (Qc), Canada, H3C 3P8.

J. Berstel,
LITP, IBP, Université Pierre et Marie Curie, Paris 6,
4 place Jussieu, 75252 Paris Cedex 05.

bergeron@lacim.uqam.ca
brlek@lacim.uqam.ca
berstel@litp.ibp.fr